# A Survey on Techniques of Secure Live Migration of Virtual Machine

Jyoti Shetty
R V College of Engineering
Department of CSE
Bangalore

Anala M R
R V College of Engineering
Department of CSE
Bangalore

Shobha G
R V College of Engineering
Department of CSE
Bangalore

## ABSTRACT
Live migration is an essential feature of virtualization that allows transfer of virtual machine from one physical server to another without interrupting the services running in virtual machine. Live migration facilitates workload balancing, fault tolerance, online system maintenance, consolidation of virtual machines etc. Unfortunately the disclosed vulnerabilities with the live migration pose significant security risks. Because of these security risks the industry is hesitant to adapt the technology for sensitive applications. This paper is an investigation of attacks on live migration of virtual machine and discusses the key proposed and implemented approaches to secure live migration.

## General Terms
Virtualization, live migration, security.

## Keywords
Live migration, virtualization, security issues, virtual machine, virtual machine monitor.

## 1. INTRODUCTION
Virtualization technology was introduced in late 1960s by IBM [1]. The expensive and powerful mainframe computers were underutilized at that time. Hence, to maximize the utilization of expensive hardware resources, techniques such as multiprogramming and timesharing were developed. These techniques formed the basis of concept of "virtualization". Virtualization is defined as the abstraction of hardware resources to facilitate better sharing of resources. Virtualization can be at different levels: application virtualization, desktop virtualization, network virtualization, storage virtualization and system virtualization etc [2]. Thus virtualization helps enterprises reducing investment and operational cost.

The term "server virtualization" or "system virtualization" means ability to run entire virtual machine, including its own operating system (Guest Operating System), on another operating system (Host Operating System). The virtual machine (VM) is defined as 'an efficient isolated duplicate of real machine' [3]. The Virtual Machine Monitor (VMM) also called hypervisor is a layer of software that emulates the hardware interface seen by the VM. The VMM completely controls system resources [3].

Live migration is essential feature of virtualization defined as a process of dynamically transferring running VMs from one physical server to another with little or zero downtime and without interrupting services running in VM[4]. Live migration is powerful tool for system administrator. It is required in many cases like online system maintenance, fault tolerance, workload balancing, testing and consolidation of VMs etc. For instance, due to resource conflict the VMs running on the same physical machine may fail to serve continuously. To avoid failover of the VMs, it becomes necessary to live migrate one or more VM running on one physical server to another physical server for continued and uninterrupted service. Live migration at present is performed manually. However research is going on for automated live migration.

Most of the commercial and open source hypervisors now support live migration. For example VMware's (VMotion), Xen, KVM (kernel based virtual machine), Oracle's Virtual box etc.

Most of the previous work has focused on the implementation of live migration with little or no consideration towards its security. Unfortunately several vulnerabilities are disclosed [5] in the implementation of live migration in Xen, VMware and etc. The major one is the migration protocol does not encrypt migration data. All migration data i.e. kernel memory, application state, sensitive data such as passwords and keys etc are transmitted as clear text. Thus there is no confidentiality of transmitted data. Other vulnerabilities migrating a VM to untrusted platforms, authentication and authorization of operations that control VM, integrity of VM data, bugs in hypervisor/migration module code etc.

A secure live migration requires

- The source and destination platforms are trusted.
- Authenticated and authorized management capabilities (VM creation, deletion, migration etc).
- The migration data should remain confidential and unmodified during the transmission.
- Mechanism to detect and report suspicious activities.

In essence the live migration process requires the typical defense-in-depth approach for it to be secured.

In following section 2 will discuss three categories of attacks on live migration of virtual machine and possible way out for each. In section 3 we will discuss four proposed and implemented approaches to secure live migration.

## 2. ATTACKS ON LIVE MIGRATION
Jon Oberheide et al has empirically demonstrated the need for secure migration process in [6]. Based on the study the attacks here are categorized on the basis of the *causes that let attack happen*. The categories of attacks are inappropriate access control policies, unprotected transmission channel and loop holes in the migration module.

## 2.1 Inappropriate access control policies

An inappropriate access control policy allows an unauthorized user to initiate, migrate and terminate a virtual machine. The access control policy also decides access to hypervisor (Domain 0), isolation between VMs on same machine and resource sharing etc. A security lax can help an attacker to perform following attacks

i. *Denial of service attack*: The unauthorized attacker can initiate large number of outgoing migrations onto a legitimate virtualized host server. Thus overloading target server, decreasing its performance or at worst disrupting service it provides. Also it is possible for an unauthorized attacker to make the VM to migrate from server to server, reducing the performance of service provided by VM.

ii. *Internal attacks:* This can be result of unauthorized attacker migrating VM with malicious code to legitimate target hypervisor. This provides a platform for malicious VM to perform internal attacks on target system. For example gaining control over the target hypervisor and other guest VMs.

iii. *Guest VM attack*: An attacker initiates a request for an incoming migration of a VM. When the requested VM is migrated the attacker gains control over the migrated VM and then performs an attack by executing a rouge code on it, crashing it etc.

iv. *False resource sharing:* An attacker system can falsely advertise available resources, influencing other VM to migrate to this compromised VM.

*v. Inter VM attack:* The VMs running on same machine can communicate with each other. If a policy is not defined for controlled communication, a malicious VM can attack other VM running on same machine.

*Way out:* To prevent an attacker from performing such an unauthorized activities appropriate access control policies (acls) must be defined. Access control policies define who can migrate out a VM, who can request to migrate in a VM, Who can suspend a VM, whether a user can terminate VM, and other such decisions. These acl's must be authenticated and resistant to tampering. These decisions are discussed in role-based policy approach in [6]. Xen provide sHype- Mandatory access control for Xen. Guidelines for configuring sHype are given in Xen user manual [7].

The acl's can be accompanied with a firewall to check that migration is from allowed source and to allowed destination systems. A firewall rule checks each packet for allowed and rejected source, destination and protocol. The specified action is taken. The action results in accepting the packet, forwarding packet or rejecting the packet.

## 2.2 Unprotected transmission channel

The insecure and unprotected transmission channel is result of the migration protocol. The migration protocol does not encrypt the data as it travels over the network, thus susceptible to active and passive attacks. An attacker can gain access to the transmission channel using techniques such as ARP/DHCP poisoning, DNS poisoning and IP/route hijacking to perform passive or active attacks [6]. Passive attacks include eavesdropping of messages for sensitive data, passwords and keys, capturing authenticated packets and replying them later. Active attacks are more serious. For example manipulating authentication services like sshd, /bin/login, pam, manipulating kernel memory like slip root kits into kernel memory etc.

*Way out:* One solution is to assign a VM or group of VM to a VLAN. The VLAN isolates migration traffic from other network traffic and defines secure transmission channel for migration data. Other solutions include encryption of migration data to provide confidentiality; integrity can be preserved using MAC, digital signatures and checksums.

## 2.3 Loop holes in migration module

Vulnerabilities in migration module are stack overflow, heap overflow and integer overflow etc. Such vulnerabilities can be exploited by an attacker to inject malicious code or even halt the process.

The virtualization software is huge and complex with large number of LOC. Xen hypervisor has about 200K LOC and XEN emulator has about 600K LOC and the Host has about 1K LOC[8]. With such a huge code, bugs tend to exist. Bug reports such as those listed in NIST's National Vulnerability Database [9] show the difficulty of shipping bug-free hypervisor code. A malicious user can exploit these bugs to attack the virtualization software. Exploiting such an attack gives the attacker the ability to obstruct or access other virtual machines and therefore breach confidentiality, integrity, and availability of the other virtual machines' code or data. The virtualization software migration code must be scrutinized thoroughly to remove such vulnerabilities. If an attacker is able to gain access to hypervisor through its migration module, then that provides platform for attack on other guest VM in the target server and to VMM itself.

*Way out:* The new release of virtualization software includes patch of such vulnerabilities. The system must be updated with the recent releases and patches to be protected from such vulnerabilities. Also secure programming methods such as type safe language must be used.

The following section will discuss in detail some major approaches to secure live migration.

# 3. APPROACHES

## 3.1 Isolating the migration traffic

One approach to secure live migration against all attacks discussed is to assign a small group of VMs or even a single VM to its own host-based Virtual LAN (VLAN). VLAN is basically a segmentation and isolation tool. The VLAN isolates migration traffic from other network traffic and defines a secure transmission channel for migration.

A major drawback of VLAN-based security approach is the growth in complexity and administrative costs as the VM population grows[10]. The complexity lies in setting up and maintaining VLANs for each VM, synchronizing VLANs configuration on virtual and physical switches, troubleshooting and fix configuration errors, manage the growth and complexity of acls as number of VM increases, ensure compatibility between physical network and virtual network security policies. With migration the things become still worse where the VM continuously move between the hosts and virtual switches.

Host-based VLANs have a security gap when more than one VM is assigned to a given VLAN. In VLAN there is no traffic monitoring and filtering mechanism thus inter-VM communication within the VLAN remains invisible.

## 3.2 Network Security Engine-Hypervisor (NSE-H)

This approach [4] is based on hypervisors included with network security engines to eradicate intrusions occurring in virtual network. NSE includes firewall, intrusion detection systems and intrusion prevention system to provide security to virtualized environment. They include intelligent packet processing capability built in them. The NSE firewall work in state full way. They maintain security context for each packet and make decisions based on security context and packet content.

The figure 1 below shows basic work flow of typical firewall provided by NSEs. There are two modules in it CTM (connection tracking module) and PMM (policy matching module).

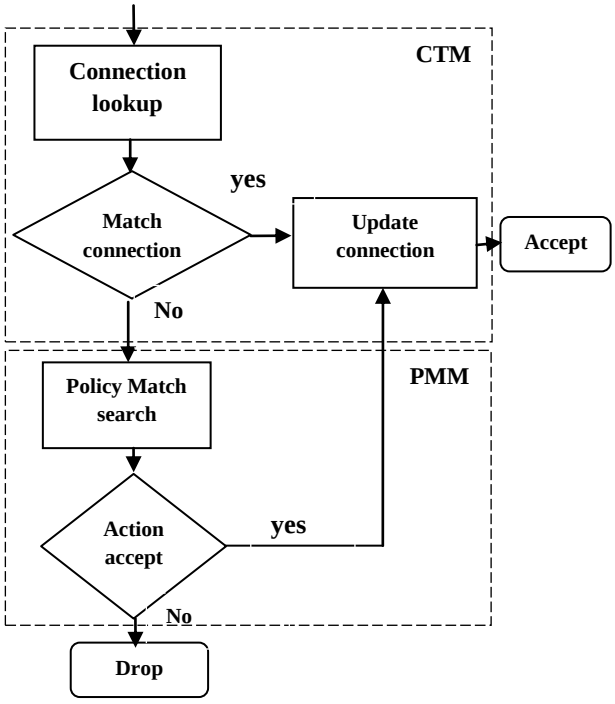


**Fig 1: State full Firewall Workflow**

The *CTM* keeps track of transport layer connection status using a hash-table like database. When a packet arrives it looks up the database based on packet header. If a match is found with the existing connection then accept action is executed otherwise the packet is forwarded to PMM for further decision.

The *PMM* stores a set of packet filtering policies defined by administrator. The filtering policies are composed of rule sets; each rule set consists of sequence of descriptors that are matched with packet content and the action to be taken.

**Live migration on NSE-H**

The problem with live migration implementation is it just encapsulates the VM execution context for transmission and not the security context. As a result at destination the VM is rejected because of missing or not matching required security context.

The solution is to include security context (SC) along with VM execution context in the migration data. The approach discussed in [4] proposes a framework for live migration on NSE-H as shown in figure 2 below. The components of architecture are VMMA, SCMA, LMC, NSE and hypervisor core.

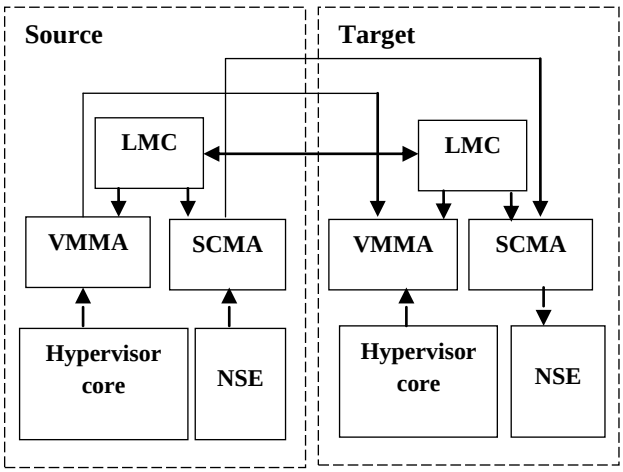


**Fig 2: CoM Framework Architecture**

*Virtual machine migration agent* (VMMA) interacts with the destination hypervisors' VMMA to transmit the VM encapsulated states to the destination hypervisor.

*Security context migration agent* (SCMA) encapsulates and sends VM related security context set through a dedicated channel.

*Live migration coordinator* (LMC) collaborates with destination hypervisors' LMC and schedules the two agents to perform migration tasks in parallel.

**Live Migration Phases**

CoM live migration extends the four phases of live migration implementation discussed in [11] as follows

***Phase I preparation*:** The LMC on source informs destination LMC to start reserving resources. Thus VMMA and SCMA both reserve the required resources and get prepared for migration.

***Phase II Iterative synchronization*:** The VMMA on source iteratively transfers the execution context of VM to be migrated to the destination. Similarly the SCMA transfers the security context of VM to be migrated.

***Phase III final synchronization*:** This phase is concerned with transfer of the recently written pages to migrated VM after the first phase of synchronization. Both the execution context and the security context are transferred by VMMA and SCMA respectively. The migrated VM is then suspended on the source hypervisor and VM related network is redirected to target server through unsolicited ARP replay adverting. The VMMA and SCMA copies the remaining execution context and security set.

***Phase IV Resumption*:** The migrated VM is resumed on target hypervisor, and the VM instance of source is discarded.

In this way the above discussed approach makes it possible for traditional security approaches like firewall, IDS etc to be effective in context of live migration.

## 3.3 Role based migration

The approach in [12] is based on use of Intel vPro and TPM hardware. The figure 3 below shows high level architecture of role based migration. It includes following modules.

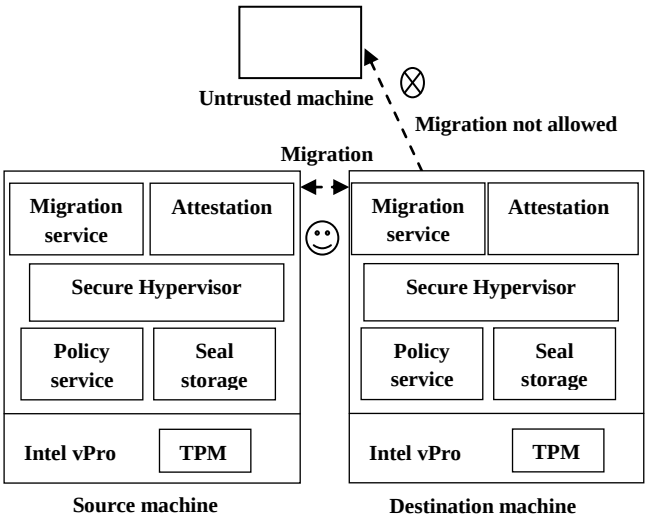


**Fig 3: Architecture of Secure Live migration**

*Attestation Service*: This module is used for attestation process. Attestation means the running hypervisor cryptographically identifies itself to a remote hypervisor to establish trust.
*Seal Storage*: This module is used to persistently store private key and role-based policies. Using the private key of the tamper resistant TPM it encrypts the data that is responsible for attestation. A hash of the booted trusted OS is also included with the encrypted data such that the TPM only allows a trusted OS with the same hash to unseal it.
*Policy Service*: This module parses and manages the role-based policies for virtual machine migration decisions, such as who has the right to migrate a virtual machine, and to which hosts this virtual machine can be migrated etc.
*Migration Service*: This is the migration module. It initiates attestation requests to remote machines to check whether the target machines meet the security requirement before migration.
*Secure Hypervisor*: This module protects the process of guest OS by Runtime Memory Measurement [13]. The detailed design and functions of secure hypervisor are shown in figure 4.

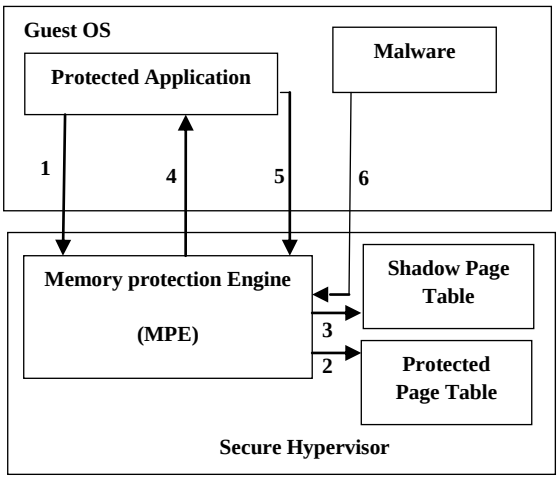


**Fig 4: Secure Hypervisor**

The description of flow[12] shown in figure 4 above is as follows.

1. Application apply a protection memory region from a hyper call registered with its manifest.
2. MPE creates corresponding protected page table
3. MPE modifies correspondent items in shadow page table
4. MPE returns the protected memory region
5. Application gets access to the protected region
6. Access to the protected region from registered application denied

The design uses remote attestation to check if target VM meets required security specification. The figure 5 below shows a detailed flowchart for the attestation process.

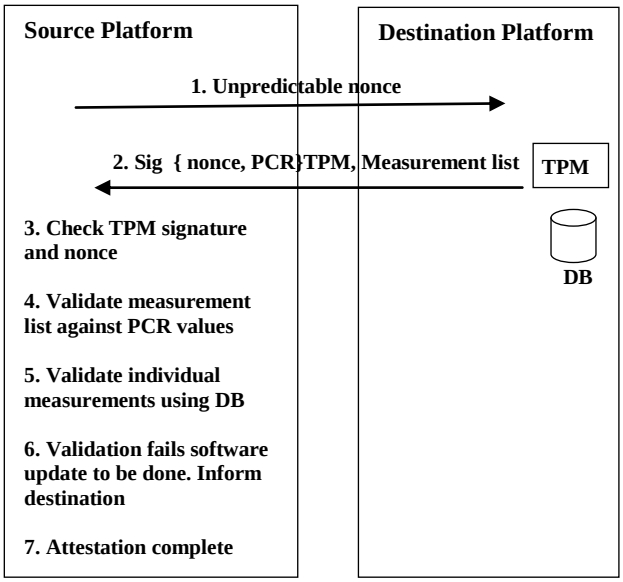


**Fig 5: Attestation process**

After successful attestation a migration session starts. Migration can be out going migration request or incoming migration request, each must satisfy the role based policies defined by the administrator. The figure 6 shows steps of role based migration.

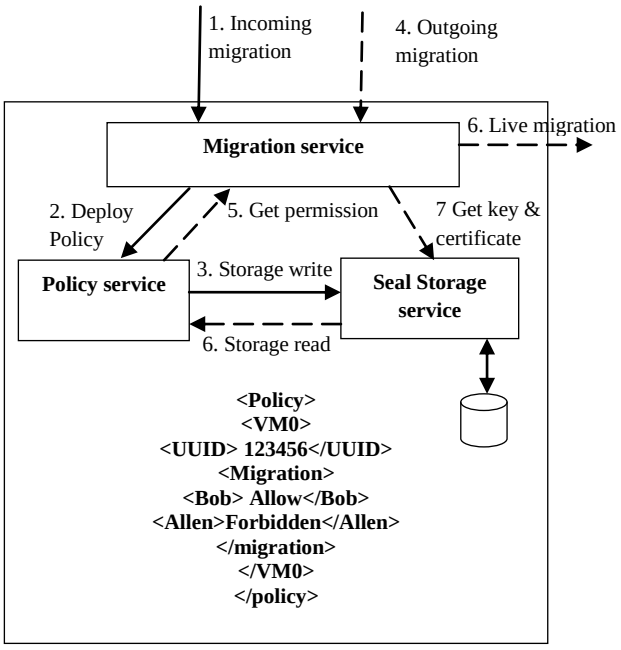


**Fig 6: Role-based live migration**

In this way the above discussed approach defines mechanism for platform attestation and a mechanism for defining role based policies for live migration. The policies are tamper resistant.

## 3.4 Virtual TPM -vTPM based migration protocol

The approach discussed in [14] is based on use of vTPM. This approach does not support live migration of VM; it supports migration of suspended VMs. The migration protocol proceeds in following phases.

*Phase I: Secure Session Establishment*

1. Transaction Layer Security (TLS) handshake protocol is used to agree on cryptographic schemes (AES, 3DES, RC4) to protect the confidentiality and integrity (SHA-1) of the data exchange that follows.
2. The source and destination mutually authenticate using public key certificates from Certificate Authority (CA) and proof of knowledge of corresponding private keys.
3. Then a pre master key is exchanged using RSA algorithm as shown in figure 7. Using premaster key, two keys $key_{enc}$ and $key_{mac}$ are derived which will be used for further encryption and integrity verification using SHA-1.

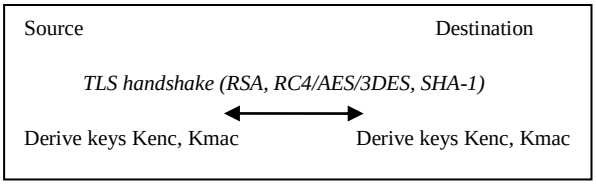


**Fig 7: Secure Session Establishment**

*Phase II: Remote attestation of the destination*

1. The source creates a new nonce (Ns1) and sends an attestation request along with it as shown in figure 8.
2. The destination includes this nonce in its signature on the PCRs related to its Trusted Computing Base (TCB). The destination also generates a nonce (Nd2) and sends it to the source which is used to ensure freshness of the VM-vTPM transfer in the next phase.
3. The source then checks the attestation reply. HMACs are used for integrity protection of the messages exchanged.
4. On verifying the integrity of the destination platform, the source locks the VM and its vTPM to prevent further changes to them. The locking mechanism is implementation specific. It then sends a SVR_ATT_OK message to the destination. If any failures occur, a SVR_ATT_FAILED is sent instead.

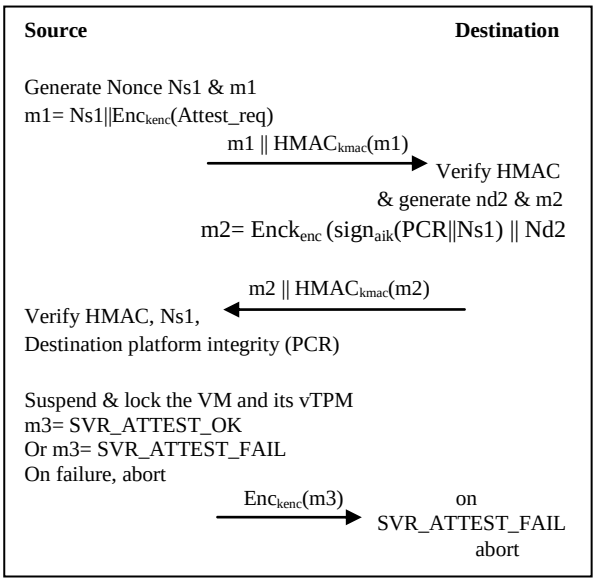


**Fig 8: Remote Attestation of Destination**

*Phase III: vTPM and VM transfer*

1. The locked VM and its vTPM are concatenated and encrypted using $K_{enc}$.
2. Then, the resulting message is concatenated with Nd1 and the corresponding HMAC is computed using key $K_{mac}$ is added.
3. The resulting encrypted data and corresponding HMAC are transferred to the destination as shown in figure 9. Nonce Nd1 is used to prevent replay of the encrypted data to the destination. Only one VM-vTPM pair is transferred per session to prevent tracking.
4. At the destination, upon receiving an encrypted VM, the HMAC is verified to ensure that the VM and vTPM were not modified during transit.
5. If the verification fails, then a negative acknowledgment (IMPORT FAILED) is sent to the source and the received VM is deleted.
6. If no modifications are detected, the VM and the vTPM are assigned their required resources. Then, the vTPM keys that are transferred in the process are imported.
7. On successful import, the destination platform sends an acknowledgment (DONE) to the source.

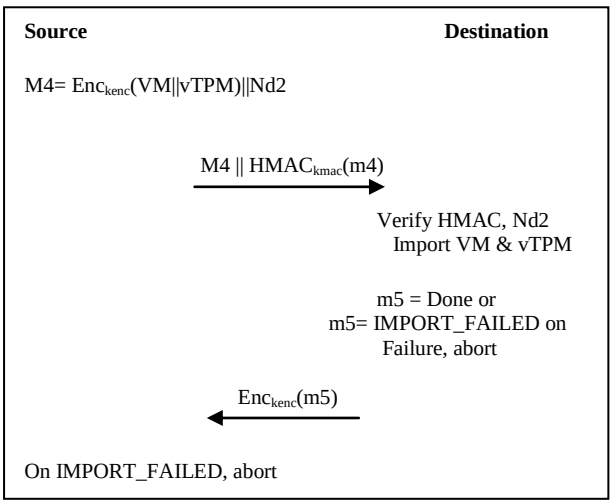


**Fig 9: vTPM and VM Transfer**

*Phase IV: Deletion at the source*

1. Upon receiving a DONE message from the destination, the source deletes the VM and vTPM as shown in figure 10.
2. However, if it receives an IMPORT FAILED message instead, it does not delete the VM or its vTPM. The source informs the destination that the migration is complete and the destination unlocks the VM and the vTPM.

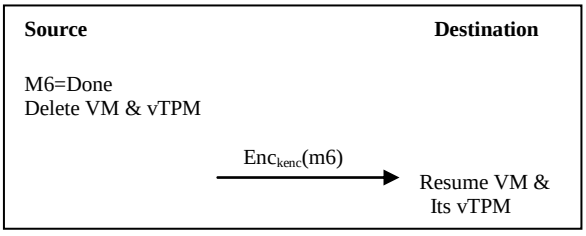


**Fig 10: Delete VM and vTPM at source**

The confidentiality of all these messages is protected using the $K_{enc}$ key.

# 4. DISSCUSSION

The approaches discussed so far address different aspects of security. Following is the interpretation of the study.

**Table 1: Interpretation of discussed approaches**

| | VLAN Appro-ach 3.1 | NSE-H Appro-ach 3.2 | Role based Appro-ach 3.3 | TPM-vTPM Appro-ach 3.4 |
|---|---|---|---|---|
| Platform Integrity Verification | | | yes | yes |
| Confidentiality and integrity of VM during migration | Depend s on VLAN Settings | | | yes |
| Authenticated and authorization of migration operation ( Access Control Policies) | Depend s on VLAN Settings | yes ( if implem ented in NSE-H) | yes | |

The *NSE-H based approach* enables the traditional security approaches like firewall, IDS, IPS present inside NSEs to work in context of live migration. It transfers the security context along with migration data so that the VM can be restored at the destination. The *policy based approach* establishes trust through attestation process, defines role based migration policies. Thus it ensures only authorized user can perform migration operations. The *vTPM based approach* provides confidentiality; authentication and trust establishment. The *vTPM based approach* is applicable only to suspend and transfer approach of migration and not to live migration.

# 5. CONCLUSION AND FUTURE WORK

To conclude no integrated approach is available that addresses Trust establishment, Confidentiality and Integrity of migration data, Authentication and authorization of migration operations which is the need of secure migration. Our future work is to develop a comprehensive framework that addresses these security aspects of live migration of virtual machines.

# 5. REFERENCES

[1] R. Adair, R. Bayles, L. Comeau, and R. Creasy, A virtual machine system for the 360/40," 1966.

[2] Mendel Rosenblum. The reincarnation of virtual machines. Queue, 2(5):34-40, 2004.

[3] Gerald J Popek and Robert P Goldberg. Formal requirements for virtualizable third generation architectures. In SOSP '73: proceedings of the fourth ACM symposium on operating system principles page 121, 1973.

[4] Chen Xianqin, Gao Xiaopeng, Wan Han, Wang Sumei, Long Xiang. Application- Transparent Live Migration for virtual machine on network security enhanced hypervisor. Research paper. China Communications. Page 32 – 42, 2011.5.

[5] Melvin Ver. Dynamic Load Balancing Based On Live Migration Of Virtual Machines: Security Threats and Effects. Thesis report Rochester Institute of Technology, B. Thomas Golisano College of Computing and Information Sciences (GCCIS), Rochester, NY, U.S.A.

[6] Jon Oberheide, Evan Cooke, Farnam Jahanian. Empirical Exploitation of live migration of virtual machines. Proc of Black Hat DC, March 24, 2008.

[7] Xen users' manual Xen v3.3. http://bits.xensource.com/Xen/docs/user.pdf

[8] Jakub Szefer, Eric Keller, Ruby B. Lee, Jennifer Rexford. Eliminating the hypervisor Attack Surface for a More Secure Cloud. In Proceedings of ACM Conference on Computer and communications Security' 2011. PP 401-412

[9] National Vulnerability Database, CVE and CCE Statistics Query Page. http://web.nvd.nist.gov/view/vuln/statistics

[10] Alternatives for Securing Virtual Networks: A Different Network Requires a Different Approach—Extending Security to the Virtual World. white paper 1000220-012-EN Dec 2011, Juniper Networks, Inc.

[11] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In Proc. of NSDI'05, pages 273–286, Berkeley, CA, USA, 2005.USENIX Association.

[12] Wei Wang†, Xiaoxin Wu, Ben Lin, Kai Miao, Xiaoyan Dang. Secured VM Live Migration in Personal Cloud. In Proceedings of 16th ACM Conference on Computer and Communications Security, CCS, 2009. ICCET, China.

[13] Dewan, P., Durham, D., Khosravi, H., Long, M., and Nagabhushan, G. A hypervisor-based system for protecting software runtime memory and persistent storage. In Proceedings of the 2008 Spring Simulation Multiconference (Ottawa, Canada, April 14 - 17, 2008).

[14] Boris Dandev, Ramya Jayram Masti, Ghassan Karame, Srdjan Capkun "Enabling Secure VM-vTPM Migration in private clouds" In Proceedings of the Annual computer Security Applications conference (ACSAC) 2011.